OPTIMIZED DESIGN OF AHB MULTIPLE MASTER/SLAVE MEMORY CONTROLLER USING VHDL

SHORT SYNOPSIS

Submitted in partial fulfillment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

to

Manav Rachna International Institute of Research and Studies
(Deemed to be University)

by

Hitanshu Saluja
Registration number: 15/Ph.D/014

Under the supervision of
Dr. Naresh Grover
Professor & Dean-Academics

FACULTY OF ENGINEERING & TECHNOLOGY

MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH & STUDIES

SECTOR-43, SURAJKUND –DELHI ROAD, FARIDABAD – 121001

FACULTY OF ENGINEERING & TECHNOLOGY

Certificate

I hereby declare that the short synopsis "**OPTIMIZED DESIGN OF AHB MULTIPLE MASTER/SLAVE MEMORY CONTROLLER USING VHDL**" being submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Electronics & Communication under faculty of engineering and technology of Manav Rachna International Institute of Research & Studies, Faridabad during the academic year 2018-19.

It is a bonafide record of original work carried out under the supervision of Dr. Naresh Grover and has not been submitted in full or part to any other university or institute for degree or diploma. Further, it is certified that short synopsis is plagiarism free and proof has been submitted.

Hitanshu Saluja
Registration number: 15/Ph.D/014

Dr. Naresh Grover
Supervisor
Dean Academics
Faculty of Engineering & Technology

(Chairman, DRC)

Chairperson, CDP

ABSTRACT

Some bus standards adopted by manufacturers to transfer the data between memory and processor or between peripherals devices due to different processing speed of processor, peripherals and memory result congestion in memory accessing time. This becomes a gridlock problem for a communication system which needs to be addressed. Since the execution of the microprocessors has been improved quickly in most recent couple of decades, processors are getting faster day by day and memories are getting bulky, it becomes mandatory to provide high-bandwidth, low-latency memory subsystem so that these processors and memories can be utilized potentially. To overcome this problem AMBA architecture introduced a new standard for SoC systems to handle high performance microcontroller system.

The Advanced Microcontroller Bus Architecture (AMBA) bus is extensively utilized in System-on-Chip (SoC) designs for the interconnection, management of functional blocks and effective communications between them. It is a very high speed and high bandwidth bus that supports multiple masters, multiple slaves and communications between them to maximize the system performance.

It has also been observed that in multiple master systems AMBA 2.0 protocol supports an arbitration technique which makes the overall system complex and perfect parallelism cannot be achieved through it because at a time only master is allowed to communicate with one slave. A single centralized bus for all functional block can also constitute a bottleneck. The proposed system will work on both AMBA 2.0 and AXI 4.0 protocols with memory controller interface. AXI 4.0 protocol supports different IDs and channel handshaking instead of arbiter. In AXI 4.0 five individual transfer channels is proposed to communicate in multiple master and multiple slave systems so that multiple masters can communicate simultaneously with multiple slaves and perfect parallelism is achieved.

For an efficient communication between a memory and IP blocks, it is proposed to design a memory controller to match the processor speed with the memory. Proposed design will operate effectively in multiple masters as well as multiple slaves model and will work as a bridge between AHB master and AHB slave.

The proposed work includes following parameters/features of AMBA:

- Burst transfer
- Several bus masters
- Single clock edge transition
- Split transaction

Proposed research work also includes development of AHB system with AHB master and AHB slave. AHB master and memory both operates on different protocol, then it will send or receive signal with the help of a interface called Slave interface, proposed research will develop slave interface as well and integration of AHB system with FIFO, RAM and ROM.

All the control structure is proposed to be designed with finite state machine and the design will be developed with VHDL language that will be further synthesized using Xilinx device. Thereafter, parameters area, speed and power will be calculated with XILINX ISE.

Keywords – Advanced Microcontroller Bus Architecture (AMBA), Advanced High Performance Bus (AHB), System on a Chip (SoC)

CONTENTS

S. No.	Description	Page No.
1	Introduction	1
2	Literature Review(Harvard Style/ APA/MLA citation)	2
	2.1 Gaps in Literature	
3	Description of Broad Area	6
4	Problem Statement & Objectives of the Study	7
	4.1Problem formulation	
	4.2 Objectives	
5	Proposed Methodology to be adopted	8
6	Expected Outcome of the Research	15
7	Proposed Time Frame	16
8	References	16

1. INTRODUCTION

With the recent development, the computing elements are being fabricated on chip which is known as “System-on-Chip”. The connections in this system-on-chip (SoCs) system are provided through buses so that the processor can communicate with memory, input-output devices and with other processors to complete the specified task. The latest research states that interconnections can also be made through network-on-chip (NoCs) instead of buses because NoCs provide more flexibility as compared to buses. Besides this research statement, some complications of NoCs stand as a question for being implemented in SoCs. For example, volume storage cannot be implemented on chip as it a separate off-chip dynamic memory. So to deal with volume in-out storage will be an issue. Since the manufacturing technique is different for memory hence, separation of memory with main system is mandatory required. To improve the execution we need to grow such effective on chip design which will be a lot quicker framework on chip arrangement and evacuates the constraint of communication architecture. One of the courses of action is "AMBA-AHB bus".

The Advanced Microcontroller Bus Architecture (AMBA) is immensely used standard design and it is registered trade mark of ARM ltd. ARM is a leading manufacturer of processors and microcontrollers for embedded systems that used in many ultra modern electronic devices i.e. media players, mobile phones and especially in SoC (System-on-Chip) designs. ARM is recognized leader for IP (Intellectual Property) standards of AMBA. All the libraries of AMBA are provided by the ARM. AMBA system has been adopted by many companies nowadays and hence it is grabbing more and more attention. Thus, it has become a standardized system for the development of SoC interconnections.

Early born buses in AMBA family were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). A short time ago high-performance AHB bus introduced by AMBA used to contain more than one bus masters or at least have a testing interface and a processor. Digital Signal Processor (DSP) or Direct Memory Access (DMA) are the commonly used AHB Master and the most commonly used AHB slaves are any internal memory and external memory interface. Other peripherals could also be used as AHB slaves but the peripherals have low bandwidth will remain on APB due to their less energy consumption and quicker response. Ensuring the integration of any current design is necessary for bridging between the existing Advanced System Bus (ASB)/Advanced Peripheral Bus (APB) designs.

The components included in an AMBA-AHB design are:

AHB Master: Bus masters are generally used for reading and writing operations with the help of address and control data. But at a particular instant only one bus master can be active and use the bus which is known as AHB master.

AHB Slave: Bus slaves are used to react read/write operation provided within an address-space range. It sends back an acknowledge signal to the bus master about the data transfer status i.e. success, failure or waiting is known as AHB slave.

AHB Arbiter: The work of arbiter is to allow only one bus master at a time to transfer the data. The arbitration protocol is generally fixed, but according to the application requirements, arbitration algorithms like fair-access or high-priority algorithms can be implemented.

AHB Decoder: The address provided for data transfer needs to be decoded. Hence, AHB decoder decodes the address and also sends a select signal to the slave which performs the transfer. For all AHB implementations, a single centralized decoder is needed.

AHB supports burst mode operation for data communication from processor to memory or other interface. It allows several words to be transferred after a single query operation, which is five to six times the normal operation. Early born buses of AMBA i.e. APB, ASB doesn't have burst feature that is why we turned to AHB. It is a bus that allows reads / writes of different sizes and supporting the burst mode as well as several masters. In case of multiple master arbitration scheme is required to distribute the access among multiple masters. AHB bus also has error signals allowing the slave to notify the master who initiated the transaction that not succeed. These two types of bus are relatively conventional each having its interest depending on the type of intellectual property (IPs) to be linked. A single centralized bus for all the functional blocks can constitute a bottleneck and restriction for performance of a system as well. Protocol AXI 4.0 of AMBA family is not providing a centred bus on a shared interconnection. AXI proposes a standard point-to-point interface between a master, slave and its interconnection. In a "crossbar" type bus each master has a dedicated or point to point connection with the slave enables simultaneous communication with each other.

2. Literature Review

Rao & Phadke (2013) proposed that since the execution of the microprocessors has been improved quickly in last few decades, it became mandatory to provide high-bandwidth, low-

latency memory subsystem so that these processors can give its full performance and can be utilized potentially. Prior to this development, there was a time span in which the development of memory subsystem and bandwidth was not as speeded as reductions in instruction execution time. To recover this mismatch, caches were being used extensively, but there were some applications which were not using caches efficiently. This results in congestion of memory accessing time. This eventually degrades system's performance and hence memory controller is designed. The data flow heading from main memory is managed by a digital circuit. This digital circuit is memory controller. It can be integrated on system's chip or it can be an individual chip. This research is about the designing of AMBA-memory controller. Designing was done on VHDL, simulated on ModelSim and the configuration was done through Xilinx on a Virtex4 family's FPGA target device.

Sharma & Ali (2013) suggested that for an effective correspondence among memory and IP block a memory controller is required to coordinate the processor speed with the memory. Author has projected the memory controller for reduction of SDRAM access latency by performing read and write operations in it based on local FIFO that would store temporary data to and from SDRAM. The ping-ponging style of writing encouraged the authors to employ two FIFOs based on the Fact that AHB bus data transfer size and SDRAM data bus width do not match with each other. In the proposed paper, Archana C. suggests a new design for memory controller with the goal of decreasing the SDRAM's latency. It uses local FIFO for storing the data temporarily.

Reddy & Kumari (2013) proposed that to match the speed of memory controller with bus architecture, S.L. Reddy constructed a memory controller which is particularly focused for SDRAM. Certain elements were incorporated into the outline which could improve the performance of the controller, such as; looking for the internal memory of the controller for the requested data the most as of late utilized data, rather than heading off to the memory to fetch it. With a specific end goal to improve the performance, SDRAMs propose features containing multiple internal banks, burst mode access and pipelining of operation executions. Getting to one bank while pre-charging different banks is empowered by the feature of multiple internal banks. This memory should be compatible with AHB system.

Arun & Kumar (2012) suggested that in various microprocessor applications, memory access time becomes a gridlock in the performance path of a system. To overcome this problem memory controller was designed. The duty of memory controller is control the memory. This

controller is generally fabricated on system's chip. The proposed paper guides to build a memory controller called as "AMBA compliant memory controller" (as AHB slave) which is compatible with AHB (AMBA). The main memory of the system consists of SRAM and ROM. While designing, some issues became the obstacle in the path of memory controller. To resolve those issues some solutions are given in reference

Ramakrishna (2012) states that though there are a lot of improvements in microprocessor technology still the memory latencies and bandwidth are improving at a lesser rate. In various microprocessor applications, memory access time becomes a gridlock in the performance path of a system. The duty of memory controller is control the memory.

Jayapraveen & Priya (2012) presented that a system's performance purely depends upon its architecture such as a computer's performance will rely upon its architecture's interconnections. The transfer of data instructions between memory and processor or between peripheral devices and memory could get choked. This becomes a gridlock problem for a communication system which needs attention. Some manufacturers have also adopted some bus standards to overcome this problem. This results in congestion of memory accessing time. To avoid this problem a memory controller is designed. The AMBA architecture introduces a new standard for SoC systems so as to handle high performance microcontroller systems.

Vijyalakshmi & Raghavaih (2012) described how to build an AMBA-AHB based memory controller that can work efficiently in multi- master and multi- slave communication model. The authors focused on the typical AMBA architecture which consists of AHB architecture having 3 masters along with 4 slaves. Structure of the SoC system with memory controller is also implemented to improve the accessing speed of the devices.

Shaikh (2012) states that though there are a lot of improvements in microprocessor technology still the memory latencies and bandwidth are improving at a lesser rate. In various microprocessor applications, memory access time becomes a gridlock in the performance path of a system. To overcome this problem memory controller was designed. The duty of memory controller is control the memory. As the speed of data fetching from memory is unable to match with the processor speed, hence we need a device which can match the speed of processor & memory.

Yueli & Ben (2011) proposed that since the performance of the microprocessors has been improved rapidly in last few decades, it became mandatory to provide high-bandwidth, low-latency memory subsystem so that these processors can give its full performance and can be utilized potentially. Prior to this development, there was a time span in which the development of memory subsystem and bandwidth was not as speeded as reductions in instruction execution time. To recover this mismatch, caches were being used extensively, but there were some applications which were not using caches efficiently. This results in congestion of memory accessing time. This eventually degrades system's performance and hence memory controller is designed. The data flow heading from main memory is managed by a digital circuit. The task assigned to memory controller is to control the memory.

Varshavishwarkama (2011) suggested some designs are larger than the usual designs which require more memory and more input-output ports. A designer can interface with external input output ports and memory through the system. To improve the performance Varshavishwarkama uses a fast data transfer in AMBA-AHB using parallel communication instead of serial communication. It likewise gives the chance to utilize masters and slaves up to 16 numbers and the data of every master is read and written at the same time. With the expansion in system frequency, it is difficult to accomplish address decoding and memory access operations in single clock cycle because of which hold up states are embedded. In any case, the technique for embeddings hold up states will bring about sensational drop in framework performance. Subsequently, the burst technique is exhibited in this paper to determine the issue.

2.1 Gaps in Literature

Some bus standards adopted by manufacturers to transfer the data between memory and processor or between peripherals devices due to different processing speed of microprocessor and memory result congestion in memory accessing time. This becomes a gridlock problem for a communication system which needs to be addressed. According to proposed research the main focus is to design AMBA-AHB-memory controller so that effective communication between multiple masters and multiple slaves taken place and multiple slave model could be achieved.

Archana C has explored the inefficient utilization of bus resources in AHB SOC. They have suggested a new design for memory controller with the goal of decreasing the SDRAM's

latency. It uses local FIFO for storing the data temporarily. This design approach ensures the data integrity but it increases the latency of system because of FIFO implementation.

S. L. Reddy depicted if an information is questioned about the latterly utilized information, searching initially occurs in the internal memory of controller as opposed to getting it to the memory. This approach creates additional complex structure for data read and write operation.

Vijayalakshmi with the point of expanding the speed of gadgets. The principle centre is around the way that how to structure this AMBA-AHB-memory controller which can work successfully as AHB-slave Peripherals. In this framework the primary memory of memory controller is SRAM and ROM.

Varsha Vishwarkama claims parallelism, the overall discussion in literature emphasis on area, speed and power for IP of AHB an AXI. The finding of literature claims an efficient architecture for low latency and area efficient design but that research is not based on latest AXI 4.0 protocol.

3 Description of Broad Area

In SOC the biggest challenge is integrity because these years the performance of the microprocessors has been improved at a higher rate as compared to the bandwidth and memory latency. This result in congestion of memory accessing time and this eventually degrades system's performance to overcome this problem memory controller highly required. Even master and it is interaction with memory operates on different protocol then it will send or receives its data with helps of its slave Interface.

It has been seen that in AXI 2.0 protocol is utilized to accomplish fast communication between processors for on chip communication. To improve the execution we need to grow such proficient on chip architecture which will be a lot quicker framework on chip arrangement and expels the impediment of correspondence design. One of the courses of action is "AHB bus". However, the ideal parallelism can't be accomplished through because single master is permitted to communicate with a single slave at a given instant. To achieve better communication or parallelism between multiple masters and multiple slaves, the proposed design having five individual transfer channels in which multiple masters can communicate with multiple slaves simultaneously. Hence, Multi Processor System on Chip (MPSOC) design achieves a perfect parallelism.

Proposed research is to develop the AHB system with multiple AHB master, multiple AHB slave and Memory Controller with their integration using VHDL language and to further synthesize it and calculate the area, power and speed. Because of multiple master requests to access the bus there is limitation in AMBA 2.0 it follows arbitration scheme, to avoid such complexity and more parallelism AXI 4.0 protocol will also be implemented along with memory controller. The developed IP shall be further synthesized and its area, power and speed will be calculated.

4 Problem Statement and Objectives of the Study

After extensive literature survey following problem statement and their proposed solutions are mentioned below

4.1 Problem Formulation

Following are the problem statement mentioned below has been formulated after doing extensive literature survey

- These years microprocessors speed has been improved at a higher rate as compared to memory latency. Due to unmatched speed of processor and memory results congestion in memory accessing time and this eventually degrades the system performance as well. To overcome this problem memory controller is highly required.
- AHB master and memory both operates on different protocols, it will send or receive signals with the help of an interface called slave interface.
- In AXI 2.0 protocol there is a limitation because single master communicates with single slave only at a given instant. Proposed design will give better parallelism between multiple masters and multiple slaves.
- In previous designs there is nothing for data and control buffering, proposed design will use FIFO.
- In protocol AXI 2.0 there is a limitation to use arbitration technique in multiple masters and multiple slave situation, which makes the overall system complex but proposed design will use AXI 4.0 protocol which supports different ID’s and channel handshaking instead of arbiter.

4.2 Objectives

As described above, the following shall be the objectives of proposed research work:

- To develop the AHB system i.e. AHB master, AHB slave and AHB slave interface using VHDL language.
- To develop the Memory controller, FIFO, RAM, ROM by using VHDL
- To integrate AHB system with memory controller using FIFO and develop the multi master situation and its control using arbiter (as in AMBA 2.0) to reduce the handshaking complexity.
- To implement the developed system along with memory controller using AXI 4.0 protocol
- To further synthesize the developed system and calculate its area, power and speed to compare the results with the contemporary work done in this area.

5. Proposed methodology to be adopted

Details of proposed methodology and work flow have been categorized in four different phases as mentioned below:

5.1 Phase-I

Development of AHB system compatible with AMBA2.0 protocol is proposed in Phase-I, as shown in figure 5.1 proposed implementation of AHB model. The data transfer is started by AHB master and is acknowledged by AHB slave. The communication is held through the slave-to-application interface. At first the master initiates the data and control signal but these control signals further cannot directly interact with memory. Therefore, data processing is carried out by slave and further forwarded through interface.

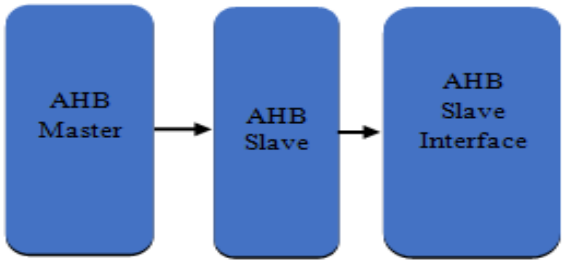

Figure. 5.1 : AHB system's top architecture

AHB master's top architecture is depicted in Figure 5.2. It comprises of controller and address generator which is responsible for generation of address depending upon the H-burst. The address is initially fetched to CPU and then that address is forwarded to multiplexer.

Now depending upon the selection it transfers its data to address generator. The various addresses generated by the address generator are administered by the counter which decides that how many addresses are needed depending upon the H-burst. Controller is implemented using finite state machine which is responsible for handling the complete operation of various blocks.

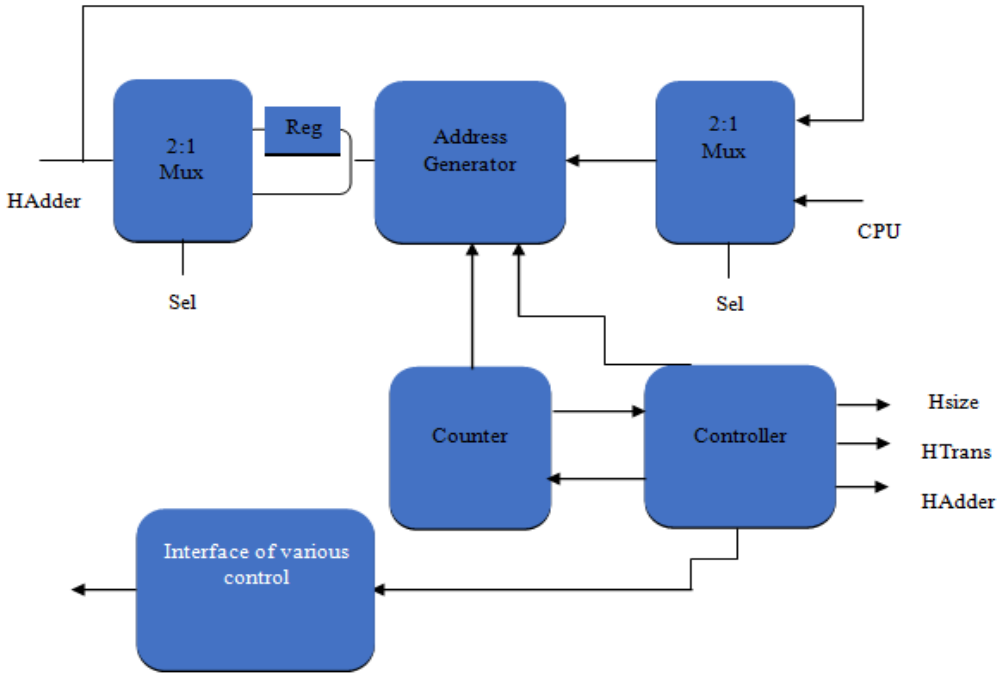

Figure. 5.2 : AHB master's top architecture

5.2 Phase-2

To develop memory controller and RAM, ROM, FIFO will be proposed in phase-2.AHB compliant's master's top architecture is depicted in Figure 5.3. In the proposed method FIFO is used for data and control buffering. This is done with the goals of assigning different clock cycles (CLK) to slave and memory which in turn gives a reliable communication and also decrease the complexity. Depending upon the READ and WRITE operations data is further communicated through RAM or ROM.

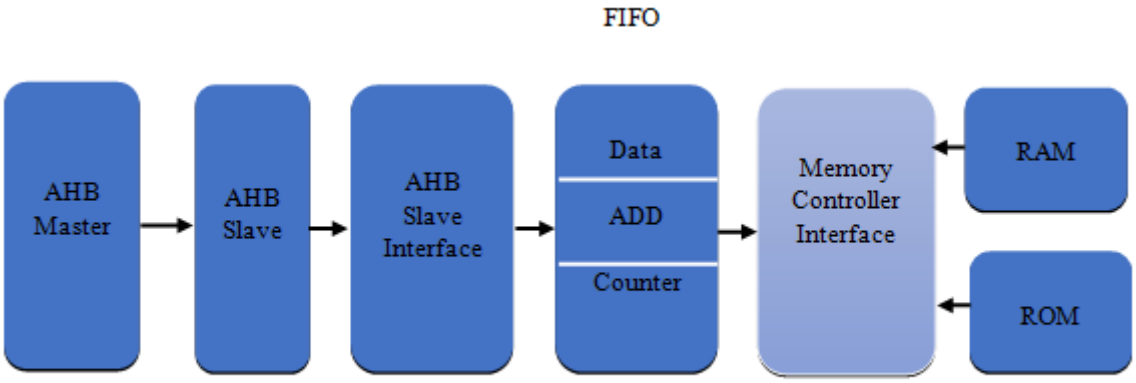

Figure 5.3 : AHB compliant's top architecture

The flow diagram of Memory Controller (MC) is depicted in Figure 5.4. This memory controller is a finite state machine implementation. Initially it is in "idle" condition that is the system "reset state" where no operation is being executed. When the START signal is received then, the FSM is triggered and the CMD state decides the further operation depending upon the instructions given. According to the Read/Write instructions, it moves to RAM READ, RAM WRITE and ROM READ operation.

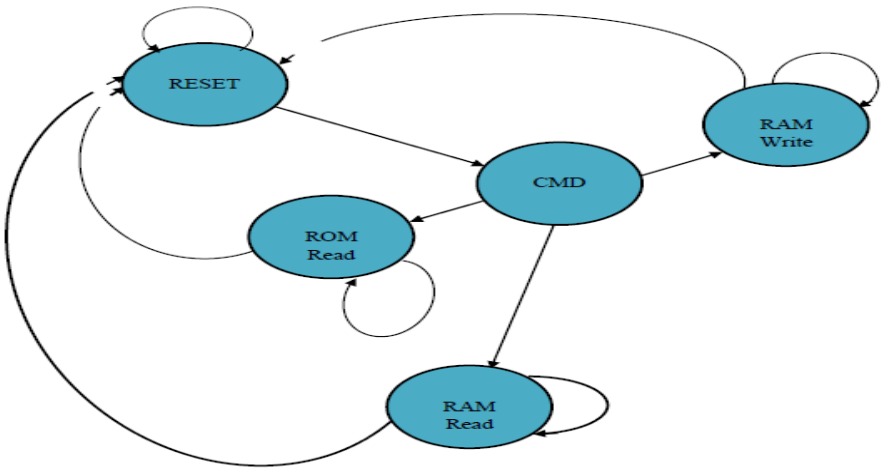

Figure 5.4 : Memory Controller (MC)

The main memory controller has to perform two main functions i.e. to tackle the time intervals between two consecutive commands and to keep the record of the rows that are

active. This activation of row consumes more time and hence the core memory controller is comprises of look-ahead function where an arbitrator has an authority to decide which instruction will take place after the current instruction is finished. There is a possibility that a row can be activated in advance only if the upcoming command is not going to access the same bank/chip which is being used by the current command.

5.3 Phase-3

Development of the AHB system (Multiple Master) with memory controller compatible with AMBA 2.0 protocol is proposed in phase-3. The topmost architecture of multiple master AHB memory controller is shown in Figure 5.5. Data is initiated by master and granted by arbiter to carry the further communication through slave-to-memory interface. If multiple processors are interacting then arbitration requires. AMBA 2.0 supports the arbiter concepts at the cost of system complexity.

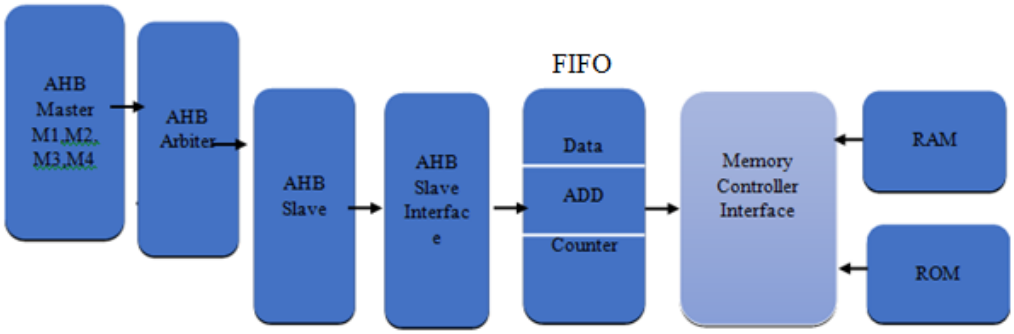

Figure 5.5 : Multiple AHB-memory controller system's topmost architecture

5.4 Phase-4

In proposed research AHB system with memory controller application compatible with AXI 4.0 protocol is proposed. In this as shown in figure 5.6 there are five different channels for communication in which three are for the writing and two for the reading purpose i.e. write address channel, write data channel, write response channel, read address channel and read data channel.

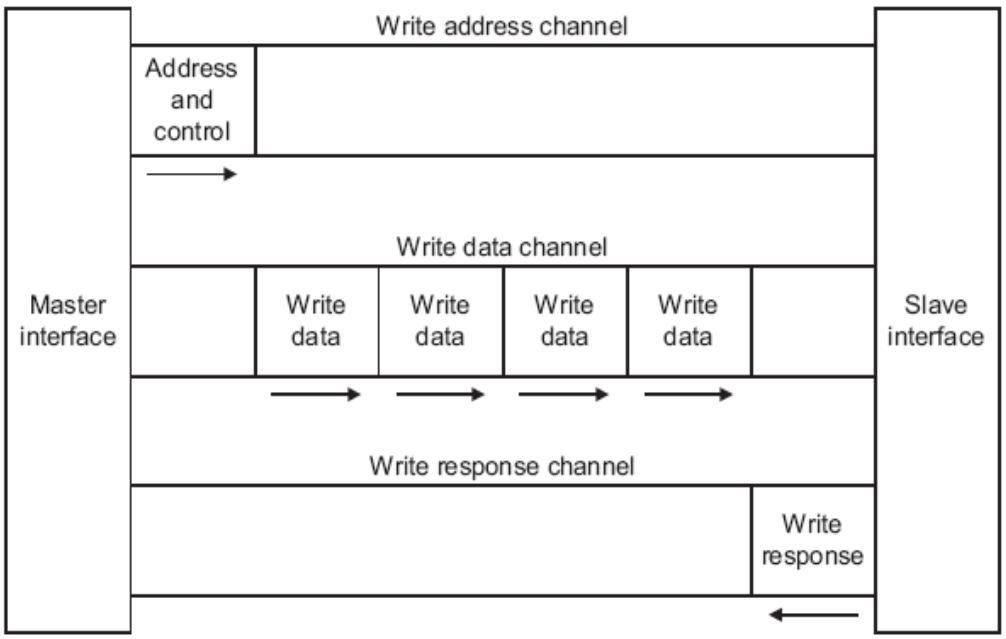

Figure 5.6: Write Address Channel [1]

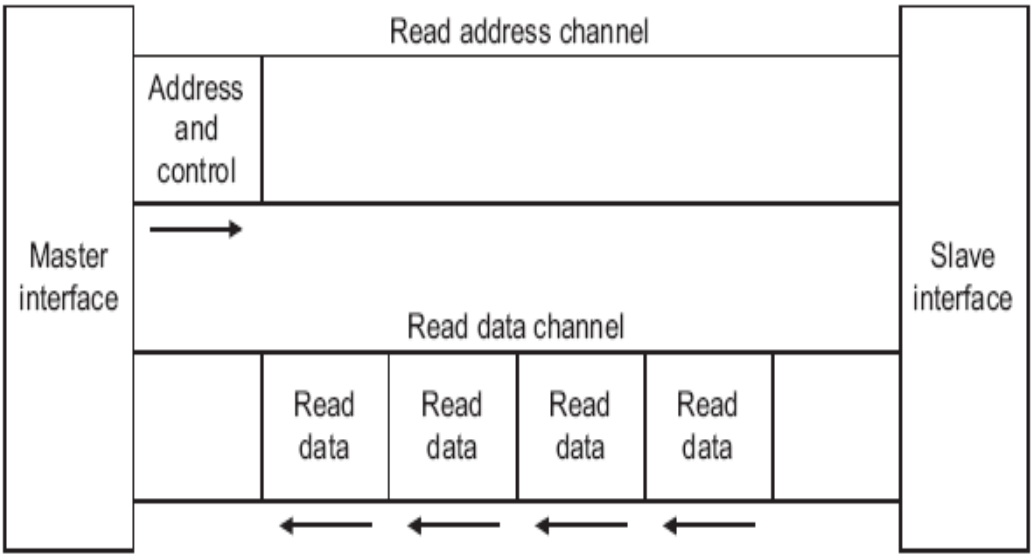

Figure 5.7: Read Address Channel [1]

The AXI protocol is burst-based. When a data needs to be transferred then its nature is described by the address channel which stores the address and control data about each execution. The data between the master and slave is communicated using a write operation to slave through the data channel and using a read operation to the master. In write operation the information flow is from master to slave but to respond back, the AXI protocol has an extra

write channel so that the slave can acknowledge the master about the completion of data transfer. The AXI protocol permits:

- The information about the address to get passed before the actual data is transferred.
- Provides assistance for multiple extraordinary executions.
- Provides assistance for out-of-order completion of executions.

We will develop the AXI 4.0 based memory controller system.

Proposed Block Diagram

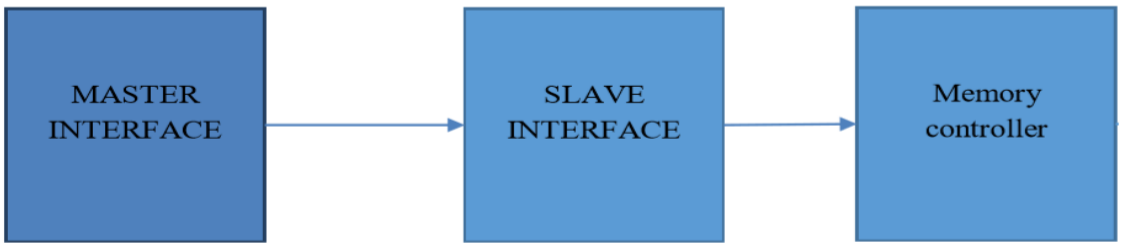

Figure 5.8 Proposed AXI master/slave interface with memory controller

Proposed Figure 5.8 shows the block based architecture of master interface of AXI protocol and slave interface with address generator. Further, slave interface is connected to memory controller for read and write operation according to read and write channel separately. This claims higher parallelism because of separate channels.

5.4.1 AXI DDR3 Block Diagram and Description

As appeared in figure 5.9 how interconnect communicates with DDR3 RAM, in the event that master needs to send a few information to the DDR3 RAM, at that point first request is exchanged to master interface which made substantial flag high to pass request of processor to exchange the demand to the interconnect and finely to the DDR3 controller. This required data of substantial flag might be reacted emphatically at slave itself by attesting the prepared flag high. Note that, as per convention the correspondence among master and slave might be built up once both prepared and substantial flag are made high and after that the correspondence between interfaces set up and information exchange will be done. Here in the diagram 3.9 the decoder has direction of choosing the DDR3 controller as one of slaves.

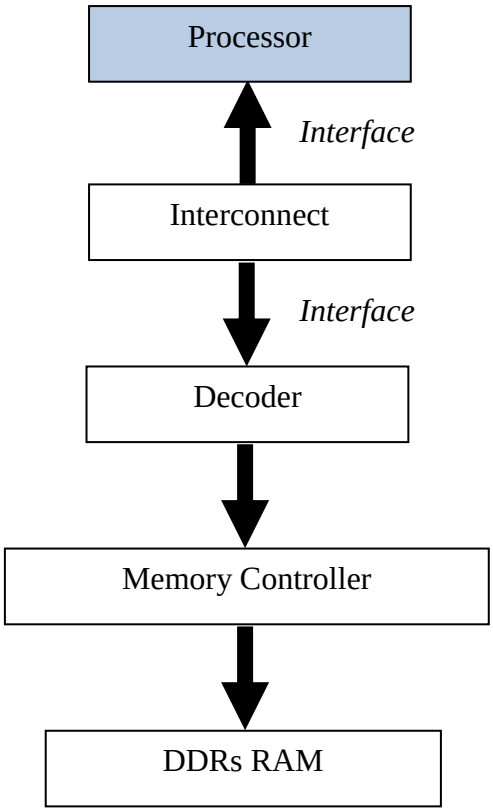

Figure 5.9: AXI Interface Protocol Interfaced with DDR3 Controller

5.4.2 Generalized Interconnect

As shown in Figure 5.10 interconnect represents the arrangement of multiple master and slave in AXI systems. Interconnect is the principle heart of this correspondence bus as it gives all the insight. It has five individual channels for exchanging the location and information.

The main features of the AXI protocol are:

• Has separate data phase, address or control commands.
• Performs burst transactions after issuance of the address.
• Advisable for projects that require high bandwidth and low latency
• Flexibility in the implementation of inter communicable architectures
• Communication with the interface of the AMBA AHB protocols and APB

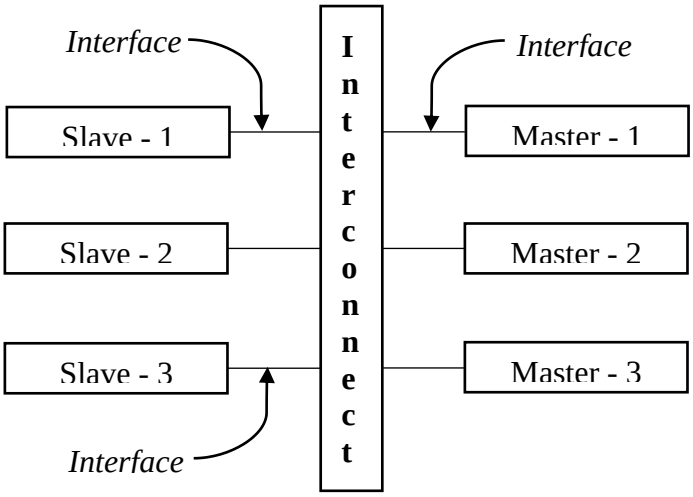

Figure 5.10: Interconnect

• Easy addition of record stages
• Has the ability to issue multiple pending addresses
• Has addressing channels, read data, read response, writing and writing response. Channels
• Addressing
• Reading data
• Reading response
• Writing data
• Writing response

6. Expected Outcome of the Research

In earlier approach there was an interface block to connect all blocks sequentially, which is having more conditional statements, hence it has created the more latches with references of four input look-up tables. The earlier approach used storage block along with finite state machine based controller, which costs additional hardware and complex handshaking. With the proposed objective of developing AMBA AHB memory controller with only one slave controller and different masters, the whole system shall be designed to achieve high throughput using pipeline approach, which shall be further synthesized using Xilinx ISE EDA tolls. It is expected that objective of high speed will be achieved. Furthermore, the proposed method is expected to reduce the hardware as compared to earlier contemporary work and reduce the device utilization summary which will give better hardware utilization than earlier approach.

7. Proposed Time Frame

Time Frame	Dec 15-June 2016		July-Dec 2016		Jan -June 2017		July-Dec 2017		Jan-June 2018		July-Dec 2018		Jan-June 2019		July-Dec 2019		Jan-June 2020		July-Dec 2020	
Months	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60
Year	1				2				3				4							
Course Work																				
Extensive Literature Survey																				
Problem Statement																				
Development of AHB system																				
Development of Memory Controller																				
Integration of AHB system with memory Controller																				
Comparison of the developed system with previous research in terms of its area, power and speed																				
Create Multiple Masters ans slaves situation with the help of arbitratio technique using AXI 2.0																				
AXI 4.0 protocol will be implemented along with memory controller																				
Developed system is further synthesized and its area, power, speed will be calculated																				
Detailed Writing and submission report																				

References

Rao, S. and Phadke, A.S., 2013. Implementation of AMBA compliant memory controller on a FPGA. *International Journal of Emerging Trends in Electrical and Electronics*, *2*, pp.20-23.

Sharma, A.C. and Ali, Z., 2013. Construct High-Speed SDRAM Memory Controller Using Multiple FIFO's for AHB Memory Slave Interface. *International Journal of Emerging Technology and Advanced Engineering*, *3*(3), pp.907-916.

Reddy, S.L. and Kumari, A.K., 2013. Architecture of An AHB Compliant SDRAM Memory Controller. *International Journal of Innovations in Engineering and Technology (IJIET), ISSN*, pp.2319-1058.

Arun, G. and Vijaykumar, T., 2012. Improving Memory Access time by Building an AMBA AHB compliant Memory Controller. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *1*(7).

Ramakrishna, S., Venugopal, K., Bhasker, B.V. and Rao, R.S.P., 2012. Hdl Implementation of Amba-Ahb Compatible Memory Controller. *International Journal Of Computational Engineering Research (ijceron-line. com)*.

Controller, M., 2012. Design of memory controller based on AMBA AHB protocol., *Elixir International Journal*, 11115-11119.

Vijayalakshmi, C. and Raghavaiah, M.B., 2012. Implementation of AMBA AHB Compliant Memory Controller with Peripherals. *International Conference on Information Technology, Electronics & Communication (ICITEC)*, 68-72.

Shaik, K., Mohiddin, M. and Zabirullah, M., A Reduced Latency Architecture for Obtaining High System Performance.', *International Journal of Recent Technology and Engineering (IJRTE)*, 1-5.

Yueli, H. and Ben, Y., 2011, January. Building an AMBA AHB compliant memory controller. In *2011 Third International Conference on Measuring Technology and Mechatronics Automation* (Vol. 1, pp. 658-661). IEEE.

Vishwarkama, V., Choubey, A. and Sahu, A., 2012. iImplementation of AMBA AHB protocol for high capacity memory management using VHDL. *International Journal on Computer Science and Engineering*, *4*(3), p.380.

Zhao, B., 2009, October. High speed DDR memory interface design. In *2009 IEEE 8th International Conference on ASIC*(pp. 11-11). IEEE.

McGee, S.W., Klenke, R.H., Aylor, J.H. and Schwab, A.J., 1994, September. Design of a processor bus interface ASIC for the stream memory controller. In *Proceedings Seventh Annual IEEE International ASIC Conference and Exhibit* (pp. 462-465). IEEE.

Datta, A. and Singhal, V., 2008, January. Formal verification of a public-domain DDR2 controller design. In *21st International Conference on VLSI Design (VLSID 2008)* (pp. 475-480). IEEE.

Goossens, K., Gangwal, O.P., Röover, J. and Niranjan, A.P., 2005. Interconnect and memory organization in SOCs for advanced set-top boxes and TV. In *Interconnect-Centric Design for Advanced SoC and NoC* (pp. 399-423). Springer, Boston, MA.

Ramagundam, S., Das, S.R., Morton, S., Biswas, S.N., Groza, V., Assaf, M.H. and Petriu, E.M., 2014, May. Design and implementation of high-performance master/slave memory controller with microcontroller bus architecture. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings* (pp. 10-15). IEEE.